National Taiwan University

2012 Final Contest

A	Amazing Magnets Puzzle	1 s
B	Beguine Dance	1 s
C	Checking Network	12 s
D	Dog Food	4 s
E	Easy In-Order Delivery	2 s
F	Fast String Coprocessor	2 s
G	Guild of Tomatoes	12 s
H	Hawk Nests	12 s
I	Illumination of Buildings	5 s
J	Joint Sword Flame	10 s
K	Key Lime Pie	2 s
L	Lazy Dreamoon	3 s

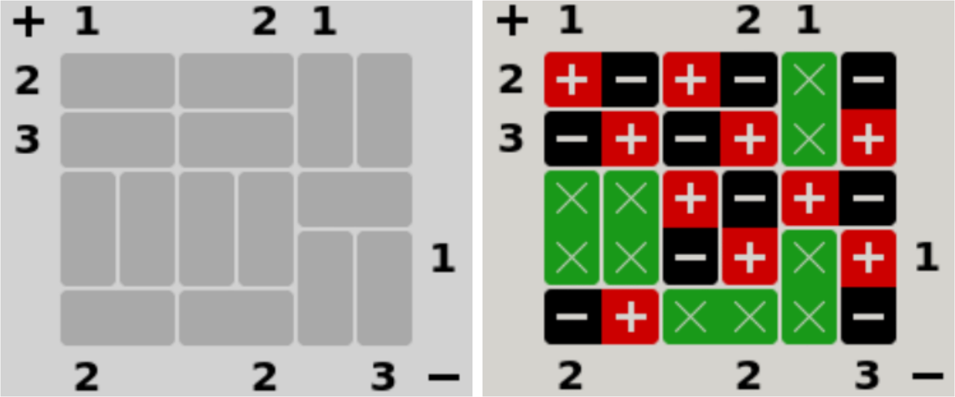
December 15, 2012

(This page was intentionally left blank.)

A. Amazing Magnets Puzzle

time limit: 1 second

The puzzle game Magnets involves placing a set of domino-shaped magnets (or electrets or other polarized objects) in a subset of slots on a board so as to satisfy a set of constraints. For example, the puzzle on the left has the solution shown on the right:



Each slot contains either a blank entry (indicated by ‘x’s), or a “magnet” with a positive and negative end. The numbers along the left and top sides show the numbers of ‘+’ squares in particular rows or columns. Those along the right and bottom show the number of ‘-’ signs in particular rows or columns. Rows and columns without a number at one or both ends are unconstrained as to the number of ‘+’ or ‘−’ signs, depending on which number is not present. In addition to fulfilling these numerical constraints, a puzzle solution must also satisfy the constraint that no two orthogonally touching squares may have the same sign (diagonally joined squares are not constrained).

Input Format

The first line of the input will contain a number T ($T \leq 1000$), indicating the nubmer of puzzle specifications.

Each specification, in free format, will consist of

- Positive integers $W \leq 7$ and $H \leq 7$, indicating the width and height of the board in squares. At least one will be even.
- Two sequences of W integers, indicating the numbers along the top (+) and bottom (−) edges respectively. Values of -1 indicate missing numbers.
- Two sequences of H integers, indicating the numbers along the left (+) and right (−) edges respectively (from top to bottom). Values of -1 again indicate missing numbers.

- *H* strings of *W* characters each, indicating the slots, or portions thereof, contained in each of the *H* rows (from top to bottom). Each string gives one row. Each character in a string is one of '`T`', '`B`', '`L`', or '`R`'. '`T`' indicates that the square is the top of a vertical slot, '`B`' the bottom of a vertical slot, '`L`' the left of a horizontal slot, and '`R`' the right of a horizontal slot.

Output Format

For each puzzle, print out a solution, using the format shown in the sample below. Leave one blank line after each printed solution.

Sample Input

```
2
6 5
1 -1 -1 2 1 -1
2 -1 -1 2 -1 3
2 3 -1 -1 -1
-1 -1 -1 1 -1
LRLRTT
LRLRBB
TTTTLR
BBBBTT
LRLRBB
3 4
2 -1 -1 -1 -1 2
-1 -1 2 -1 0 -1 -1 -1
TTT BBB TLR BLR
```

Sample Output

```
+-+-x-
-+-+x+
xx+-+-
xx-+x+
-+xxx-

+x+
-x-
+-+
-+-
```

B. Beguine Dance

time limit: 1 second

Tmt514 is very slim, and so are his brothers Tmt001, Tmt002, $\cdots$, and Tmt999. One day, they would like to do a talent show at school to demonstrate their slimness and smartness. Their plan is the following:

Initially, n of Tmt brothers line up according to their ID numbers. Since any one of them is as smart and slim as another, let's just assume they are Tmt001 $\cdots$ Tmt-n. Then, they request the audience for two permutations of 1 to n, namely A and B. After that, Tmt brothers will immediately design two dancing steps P and Q, which are also permutations, satisfying the following requirements: When they perform P and then Q, they will end up with permutation A; when they perform Q and then P, they will end up with permutation B. Note that after performing P, the one at the i-th position arrive position $P(i)$, similar for Q. Also, ending up with permutation A means that Tmt-i should stand at position $A(i)$ in the end, similar for B.

To demonstrate that you are as smart as Tmt brothers, please write a program to compute P and Q from A and B.

Input Format

The first line of the input file contains T ($T \leq 100$), the number of test cases.

Each test case is described in three lines. The first line contains an integer n ($1 \leq n \leq 1000$), the number of Tmt brothers doing the talent show. The second line contains n integers, denoting the permutation A. The third line also contains n integers, denoting the permutation B. It is guaranteed that A and B are valid permutations of 1 to n.

Output Format

For each test case, please output "`No`" if there is no such P and Q exist. Otherwise, please output "`Yes`" on one line, $P(1) \cdots P(n)$ on the next line, and then $Q(1) \cdots Q(n)$ on the third line. The numbers should be separated by a white space.

If there are multiple solutions, you may output any of them.

Sample Input

```
2
3
1 3 2
2 1 3
3
1 2 3
3 2 1
```

Sample Output

```
Yes
3 1 2
3 2 1
No
```

C. Checking Network

time limit: 12 seconds

Tmt514 is an networking engineer (also the 514th employee, if you would like to know) in Taiwan Mass Transport (TMT) company, which maintains the subway system in Taiwan. (Yes, in 2050 every city in Taiwan are connected by the TMT subway system.) Tmt514's job is to analyze and optimize the network connection performance in the transport system.

The system has n stations, and m lines each connecting two stations. Besides selling ordinary single-journey tickets, TMT also sells three-way round-trip tickets at very low prices. That is, a passenger may buy tickets $x \to y$, $y \to z$, and $z \to x$ at the same time for some different stations x, y, and z that have lines connecting any two of them. Such round-trip tickets are very beneficial for passenger like Ferng, who lives at Xinhai, works at Taipei City Hall, and has dinner at Zhongxiao Fuxing everyday. (Yes, in 2050 any two of these three stations are connected directly.)

Since Tmt514 is so smart, soon he gets bored of the job. Besides analyzing the network, now he is more interested in the *net work* done by the transport system on each line. For a line connecting stations x and y, if a passenger goes from x to y via this line, it is counted as 1 unit of work on this direction. If another passenger goes from y to x, it is counted as 1 unit of work on the opposite direction. These 2 units of work at opposite directions can be cancelled out, so the net work on this line is 0.

Tmt514 has the records of the net work on each line. He is wondering if the net work can be achieved by passengers using three-way round-trip tickets only. Furthermore, he is also interested in the generalized question: Given that for every station the number of incoming and outgoing passengers are the same, can all the possible net works be achieved by passengers using three-way round-trip tickets only?

To demonstrate that you are smarter than Tmt514, you have to answer both questions before him.

Input Format

The first line of the input file contains T ($T \leq 100$), the number of test cases.

Each test case begins with a line containing two integers, n and m ($n \leq 10^5$, $m \leq 2n$), denoting the number of stations and lines, respectively. Them m lines follow. The i-th line contains three integers, x_i, y_i, and v_i ($1 \leq x_i, y_i \leq n$, $|v_i| \leq 100$), where x_i and y_i are the stations that the i-th line connecting, and v_i is the net work record on this line. $v_i > 0$ means that the net work is on the direction from x_i to y_i, and $v_i < 0$ means the opposite.

The input is guaranteed that no line connects one station to itself, and for any two lines they may share at most one station (regardless of the position in the input). Because of the limitation on the depth of the stations, each station can be connected by at most 4 lines.

Output Format

For each test case, please output two answers on one line separated by a white space. The first answer is “Yes” if all the possible net works (i.e. $\forall(v_1, \cdots, v_m) \in \mathbb{Z}^m$) satisfying the constraint stated above can be achieved by three-way round-trip passengers only, or “No” otherwise. The second answer is “Yes” if the net work $(v_1, \cdots, v_m)$ given in the input can be achieved by three-way round-trip passengers only, or “No” otherwise.

Sample Input

```
2
5 8
1 2 0
1 3 0
1 4 0
1 5 0
2 3 1
3 4 1
4 5 1
5 2 1
4 4
1 2 1
2 3 1
3 4 1
4 1 1
```

Sample Output

```
Yes Yes
No No
```

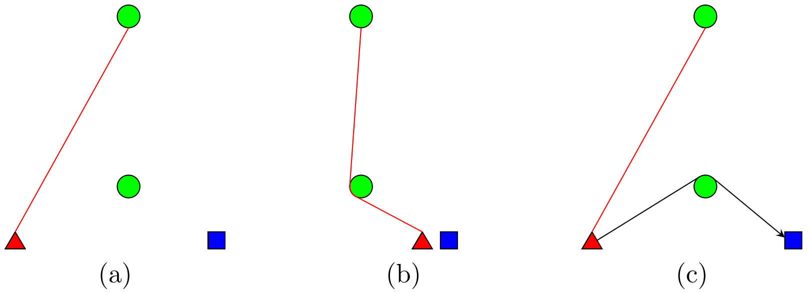
D. Dog Food

time limit: 4 seconds

Tmt514 has a smart dog, just as smart as he is. One day, Tmt514 would like to demonstrate how smart his dog is, so he set up the following game:

The game is played on a vast grassland with some coconut trees planted. Tmt514 tied his dog with a rope to the tree at $(0, 0)$. Besides this tree, there are n trees planted at $(x_1, y_1), \cdots, (x_n, y_n)$. In the initial state, the dog is at (x_D, y_D), and the rope is taut. That is, the length of the rope is equal to the distance between $(0, 0)$ and (x_D, y_D). Tmt514 has placed a piece of dog food at (x_F, y_F), which is the goal of the dog.

The thickness of the rope, as well as the width of the trees, can be ignored. However, the rope cannot penetrate any tree. In Figure 1a, the circles are trees, the triangle is the dog and the square is the goal. If the dog runs toward the goal directly, it cannot reach the goal because the rope will be blocked by the tree in the middle, as shown in Figure 1b. Instead, the dog has to bypass the middle tree as in Figure 1c.



Since Tmt514's dog is so smart, it always takes the shortest path to reach the goal. Tmt514 himself is also smart, so he knows the distance of such shortest path immediately. To show your smartness, you have to figure out that shortest distance.

Input Format

The first line of the input contains an integer T ($T \leq 250$), indicating the number of test cases.

Each test case starts with a line containing an integer n, representing the number of trees (excluding the $(0, 0)$ one). The next line contains x_D, y_D, the initial position of the dog. Yet the next line contains x_F, y_F, the position of the dog food. Then n lines follow. The i-th line contains x_i, y_i, the position of the i-th tree.

For all the test cases, the following constraints are satisfied:

- $1 \leq n \leq 8$, $-100 \leq x_D, y_D, x_F, y_F, x_i, y_i \leq 100$, and all coordinates are integers.
- No tree lies on the line segment connecting $(0, 0)$ and (x_D, y_D).
- $(0, 0)$, (x_D, y_D), (x_F, y_F), (x_1, y_1), $\cdots$, (x_n, y_n) are all distinct.

Also you may assume that if (x_D, y_D) is changed by $\epsilon < 10^{-5}$ in the direction of the origin, the result is changed by no more than 5×10^{-4}.

Output Format

For each test case, please output −1 if the dog cannot reach the goal. Otherwise, please output the shortest distance with no more than 0.001 absolute or relative error.

Sample Input

```
5
1
-4 -8
3 -8
0 -6
1
0 6
4 0
1 4
1
4 0
0 6
1 4
4
95 0
0 90
55 64
33 31
5 4
15 43
8
100 100
99 -100
60 50
6 5
12 10
24 20
30 0
70 0
-30 -10
-90 -30
```

Sample Output

```
8.0776872
7.2360679
-1
140.2870005
273.9090890
```

E. Easy In-Order Delivery

time limit: 2 seconds

Tmt514 owns an express delivery company in the United States of Tomato (UST). The service of his company is famous for its fast and reliable delivery. The main reason of its success is that the delivermen only take secure roads to deliver the packages. There are n cities in the UST, and Tmt514 has identified n secure roads to connect all the cities.

One day, a customer complained that his package was delivered slower than others'. He sent his package on Monday and got it delivered on Friday, while his friend who sent package on Tuesday got delivered on Thursday. After detailed investigation, Tmt514 found out the reason: Two delivermen took different paths to deliver packages, and the one who went first was trapped in a traffic jam.

To eradicate this problem, Tmt514 has decided to remove one secure road such that there is only one secure route from any city to another. In this way, the deliverman who goes early will also arrive early. Tmt514 is so nice that the word "overtake" is not in his dictionary.

The only thing that remains to determine is which road to remove. To demonstrate your smartness (so that Tmt514 may choose you to be his successor), you are responsible to give the best road to remove, in terms of minimizing the maximum distance between any two cities.

Input Format

The first line of the input file contains T ($T \leq 100$), the number of test cases.

Each test case starts with one line containing an integer n ($3 \leq n \leq 10^5$), the number of cities. Each of the following n lines contains two integers, x and y ($0 \leq x, y < n$), describing that there is a secure road between city x and city y. The roads are bi-directional.

It is guaranteed that there is at most one road between any two cities, and one can go from any city to another following the secure roads. The sum of n in the input does not exceed 1.5×10^6.

Output Format

For each test case, please output one line containing an integer indicating the smallest maximum distance between any two cities after removing one road.

Sample Input

```
2
3
0 1
1 2
2 0
4
0 1
1 2
2 3
3 1
```

Sample Output

```
2
2
```

F. Fast String Coprocessor

time limit: 2 seconds

The famous microprocessor company, OUTEL, has designed a new coprocessor for handling strings. You may feed the coprocessor with an input string and a program that describes the processing commands, and then it will return to you the resulting string in one clock cycle!

The input string of the coprocessor should contain 2^k lowercase Latin characters (for non-negative integer k), and the program should contain exactly $2^k - 1$ commands, where each command can be either `0` or `1`. If the input string is of length 1, the resulting string is just the same as the input. If the input string is longer, it does the following:

First, it divides the input string into two substrings of equal length, and the program – in three parts. The first two parts of the program have equal length, while the third part contains the last one command only. Then it process the first substring with the first part of the program and the second substring with the second part, obtaining the first and the second sub-results, respectively. After that, if the third part of the program is `0`, the second sub-result is appended to the right of the first; otherwise, the first sub-result is appended to the right of the second. The concatenation of the sub-results is the final result.

For example, if the input string is “`abcd`” and the program is “`0 1 1`”, the result will be “`dcab`”.

You are the quality assurance engineer of this product, so now you need to learn how to write programs for this coprocessor. Specifically, you would like to write a program that converts a string S to T, or to determine that no such program exist.

Input Format

The first line of the input contains an integer indicating the number of test cases.

Each test case consists of two lines. The first line contains the string S, and the second line contains the string T. Both strings consist of lowercase Latin letters only, and they are in the same length. The length of each string is guaranteed to be 2^k, where $1 \leq k \leq 16$. The total length of all S in the input does not exceed 3000000.

Output Format

For each test case please output “`Yes`” on one line if there is a program that converts S to T, or “`No`” otherwise. If the answer is “`Yes`”, please output $2^k - 1$ numbers on the next line, separated by a white space – a program that makes the coprocessor convert S to T. If there are multiple solutions, you may output any.

Sample Input

```
2
abacabab
baababca
abacabab
bbaaabca
```

Sample Output

```
Yes
0 1 0 1 0 0 1
No
```

G. Guild of Tomatoes

time limit: 12 seconds

Guild is a kind of organization for the pursuit of a common goal. Guild of tomatoes, of course, is for protecting tomato rights.

One day, a tomato were bullied by an apple and a mango in a juice factory. They laughed at the poor tomato because they have smartphones but he does not. By knowing this, the guild of tomatoes decides to declare a war against them to protect the fundamental tomato right.

There are n tomatoes going on the war, each with a not-necessarily-different sizes. The commander, the 514th tomato Tmt514, who is also the smartest, would like to split them into k groups and to send each group to a battlefield. But there are two problems. For maximum agility, the tomatoes in the same group should be at the same size. And for maximum strength, the sum of the unit size over k groups should be as large as possible.

It is too hard for a small tomato to grow up in a short time. Instead, Tmt514 has invented a great method for shrinking big tomatoes. Let $s_1, s_2, \cdots, s_m$ be the sizes of tomatoes in one group. Tmt514 looks at all the pairs of the tomatoes: $(s_1, s_2), (s_1, s_3), \cdots$. Each time he encounters a pair of the tomatoes (s_i, s_j) with different sizes, say $s_i > s_j$, he squeezes s_i by s_j, so that the new size of tomato i is $s_i - s_j$. This process may take a number of iterations. In the end, all the tomatoes in this group will have the same size and are ready to send to the battlefield.

Now the agility problem is solved, but the maximum strength problem remains for you. Given n, k, and $s_1, \cdots, s_n$, please arrange the n tomatoes to k groups, so that after applying the shrinking method above, the sum of the size of k tomatoes (one in each group) is maximized.

Input Format

The first line of the input file contains T ($T \leq 100$), the number of test cases.

Each test case is described in two lines. The first line contains two integers n and k ($k \leq n \leq 10^5$). The second line contains n integers $s_1, s_2, \cdots, s_n$ ($1 \leq s_i \leq 10^9$).

Note: The input is generated for this problem specifically, so s_{514} in the input may not reflect the true size of Tmt514.

Output Format

For each test case please output one line containing the maximum sum of sizes.

Sample Input

```
2
5 1
3 6 9 12 15
5 3
3 3 3 7 6
```

Sample Output

```
3
16
```

H. Hawk Nests

time limit: 12 seconds

Winter is coming! A group of hawks would like to migrate to tropical area so that they can have warm sunshine. The leading hawk has selected a big tree as there new home, and they are going to build their nests in this tree.

Like a clever rabbit has three burrows, the hawks know that they should not put all their eggs in one basket. To spread the risk of being attacked by their predators (e.g. raccoons and foxes), they are going to build 5 nests on this tree. Also, they would like the 5 nests to be as far from each other as possible, so that it takes much time for a predator to steal their eggs.

The tree consists of nodes and edges. An edge connects two nodes, and there is only one path from one node to another. The hawks are going to choose 5 nodes (not necessarily distinct) to build the nests. Furthermore, the hawks are so clever to know that their predators move in a non-constant pace on the tree. Let $t(k)$ be the time which the predators need to move along k edges. Because the raccoons and foxes can jump over the branches, the function satisfies the following criteria:

- $t(k_2) \geq t(k_1),\ \forall k_1 < k_2$
- $t(k_2) \geq \dfrac{(k_2 - k_1) \cdot t(k_3) + (k_3 - k_2) \cdot t(k_1)}{k_3 - k_1},\ \forall k_1 < k_2 < k_3$

Given the structure of the tree and the function $t(\cdot)$, please help them to find the places for the 5 nests that maximize the sum (over all pairs of nests) of the time that the predators need to go from one nest to another.

Input Format

The first line of the input file contains T ($T \leq 50$), the number of test cases.

Each test case starts with a line containing an integer n ($1 \leq n \leq 50000$), the number of nodes in the tree. The next line contains $n-1$ integers $t(1), t(2), \cdots, t(n-1)$ which are the evaluation of $t(\cdot)$ at $n-1$ possible distances ($t(i) \leq 10^8$). Note that $t(0)$ is always 0. Then $n-1$ lines follow, each has two integers x, y ($0 \leq x, y < n$), denoting an edge between node x and node y.

In 90% of the input the n is at most 2000.

Output Format

For each test case, please the maximum sum of the time as required on one line.

Sample Input

```
2
5
5 10 14 18
0 1
1 2
2 3
3 4
5
1 2 3 4
0 1
1 2
2 3
2 4
```

Sample Output

```
112
22
```

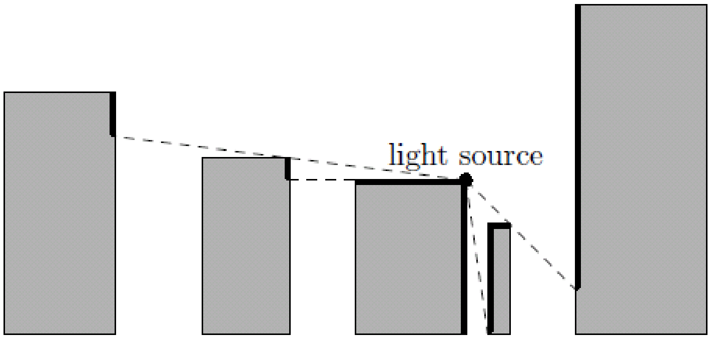
I. Illumination of Buildings

time limit: 5 seconds

The city of Harbin is famous for the nighttime illumination of its buildings. Unfortunately, the economic crisis of the world has not left the welfare of the city undisturbed. An audit performed by the city council has revealed that lighting is the single largest expense in the budget of the city. Accordingly, it was decided to cut the costs for lightning as much as possible, but without sacrificing the quality of the illumination, as the council has no desire to damage the world fame of the city.

Let's consider a 2-dimensional model of the city where each building is described by three integers L, R, H and modeled as a rectangle with edges parallel to the coordinate axes and two opposing corners at the points $(L, 0)$ and (R, H). It may be assumed that the rectangles in the model do not intersect and even do not touch each other. Line segments $[(L, 0); (L, H)]$ and $[(R, 0); (R, H)]$ are called the side edges and line segment $[(L, H); (R, H)]$ the top edge.

The city council plans to install a number of light sources to illuminate the buildings. Each light source is to be installed on a top edge of a building (possibly on an endpoint of the top edge). There may be any number of light sources on one building. It is known that a light source installed at (x_1, y_1) will illuminate all points (x_2, y_2) where the line segment $[(x_1, y_1); (x_2, y_2)]$ does not contain any internal points of any buildings. It is allowed for the segment to contain any (even infinite) number of edge and corner points of buildings.



The figure above shows a light source and all points illuminated by it.

You are asked to install the minimal number of light sources to ensure that both sides of each building are completely illuminated. A side of a building is completely illuminated if for every point P on the side (including endpoints) there exists at least one light source L that illuminates the point P.

Input Format

The first line of the input file contains T ($T \leq 150000$), the number of test cases. The line is followed by T blocks, each describing a test case.

The first line of a block contains N ($1 \leq N \leq 1000$), the number of buildings in the city. Each of the following N lines describes one building and contains three integers L, R and H ($1 \leq L \leq R \leq 10000$, $1 \leq H \leq 10000$).

It may be assumed that the sum of squares of values of N over all test cases in an input file does not exceed 1.5×10^9.

Output Format

The output file should contain one line for each test case given in the input file. Each line should contain a single integer, the minimal number of light sources required to illuminate both side edges of all the buildings in the city.

Sample Input

```
2
4
3 4 1
5 6 1
7 8 1
1 2 10
6
3 4 1
5 6 1
7 8 1
1 2 10
11 12 10
9 10 1
```

Sample Output

```
5
4
```

J. Joint Sword Flame

time limit: 10 seconds

Have you ever heard about the famous virtual reality game “Sword Art Online” (SAO)? The world of the game takes the form of a giant floating castle called Aincrad, with 100 floors in it. Each floor has a medieval-themed setting and a dungeon with a boss, which has to be defeated before players can advance to the next higher floor. The players logged into the game has to defeat the final boss at the 100th floor before they can log out. If they fail, their bodies in the real world will also die.

The main weapons used in the game are swords. And the system has specified some sword tracks so that once a player brandishes his sword following a track, a sword flame is casted toward the same direction. The enemy is damaged if it is hit by the sword flame. Different tracks cause different kinds of sword flame.

One day, Kirito, the leading role in SAO, has figured out the dual sword technique, which allows him to use two swords simultaneously. Furthermore, if the two sword flames (one from each sword) hit an enemy at the same position, the damage is tripled! Now Kirito is very interested in the dual sword technique, so he would like to know how many ways to use this technique to attack an enemy.

For simplicity, assume the world is a 2-D plane, and the enemy stands at a rectangle with sides parallel to the axes. Each sword flame (casted from a sword track) is a straight line in this plane, and the enemy is damaged if the line intersects with the rectangle. The triple damage of the dual sword technique happens when the two lines intersect in the enemy’s rectangle or on its border.

Input Format

The first line of the input file contains T ($T \leq 50$), the number of test cases.

Each test case starts with a line containing four integers x_{E1} y_{E1} x_{E2} y_{E2}, indicating that the enemy is at rectangle $(x_{E1}, y_{E1}) - (x_{E2}, y_{E2})$. The next line contains an integer n ($n \leq 100000$), indicating the number of system-specified sword tracks. Then n lines follow. The i-th line contains four integers, x_{i1} y_{i1} x_{i2} y_{i2}, indicating that the i-th track is the line segment connecting (x_{i1}, y_{i1}) and (x_{i2}, y_{i2}), and the sword flame of this track is the line connecting these two points.

It is guaranteed that no two sword flames coincide with each other. All the coordinates in the input are integers and have absolute value at most 10^8.

Output Format

For each test case, please output an integer indicating the number of pairs of sword flames that cause triple damage when using the dual sword technique.

Sample Input

```
2
0 0 10 10
2
0 0 10 20
0 20 10 0
0 0 10 10
3
-5 5 10 10
5 -5 10 10
0 5 5 0
```

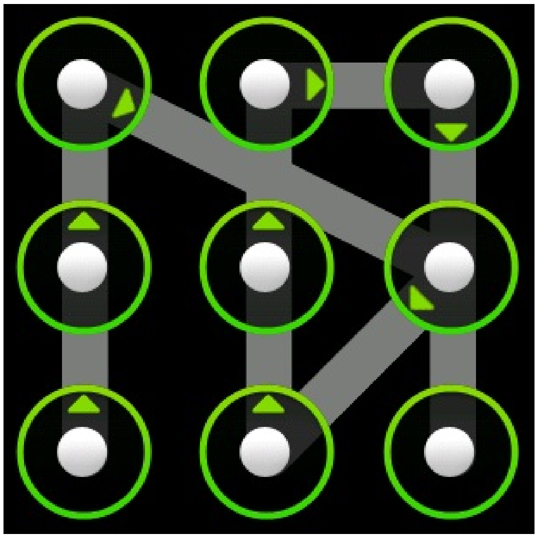
Sample Output

```
1
1
```

K. Key Lime Pie

time limit: 2 seconds

You may be familiar with the following pattern lock on Android phones. There are 3×3 nodes on the screen, and a *pattern* is a path that connects some of the nodes. If the path you draw is the same as the lock pattern set on the phone, then the screen is unlocked.



In this problem we only consider the patterns which visit all the nodes. The corresponding path of the pattern above is $(2,0) \rightarrow (1,0) \rightarrow (0,0) \rightarrow (1,2) \rightarrow (2,1) \rightarrow (1,1) \rightarrow (0,1) \rightarrow (0,2) \rightarrow (2,2)$. Note that if a node is visited multiple times, we only record the first time on the path. (See $(1,2)$ in the above example.) Moreover, for any two consecutive nodes u and v on the path, every node which lies in the straight line segment $u--v$ should be visited before u and v. (Thus, $(0,0) \rightarrow (2,2) \rightarrow (1,1)$ is not a valid path.)

Now consider the next generation of Android phones – Android Key Lime Pie (KLP). To comply with the tall-and-thin fashion of smartphones, Android KLP supports another mode of pattern lock. Instead of 3×3 nodes, it shows $n \times 2$ nodes!

Patterns over $n \times 2$ nodes look cool, but are they secure enough? Please find out how many valid patterns exist.

Input Format

The first line of the input file contains T ($T \leq 5$), the number of test sets.

Each test set starts with a line containing two integers q and p ($1 \leq q \leq 2000$, $1 \leq p \leq 2^{31} - 1$). Then q queries follow, each at one line. Each query contains an integer n ($1 \leq n \leq 2000$) as described in the problem statement.

Output Format

For each query in each test set, please output an integer on one line: the number of valid patterns using all $n \times 2$ nodes, taken modulo p.

Sample Input

```
1
2 1000000007
1
2
```

Sample Output

```
2
24
```

L. Lazy Dreamoon

time limit: 3 seconds

Dreamoon just got a new computer with Windows operating system. Since an online contest starts very soon, Dreamoon has little time to set up the environment. Usually he uses `vim` as editor and `gcc` as compiler. But this time he can only set up `gcc`, so he use the built-in `notepad` as editor instead.

The functionality of `notepad` is very limited. Unlike that one can type `5j` in `vim` to move down 5 lines, in `notepad` one have to press the down arrow key 5 times. In particular, `notepad` only offers the following ways to move your cursor.

- left/right arrow key: Move left/right one character on the current line. No effect if the cursor is already at the beginning/end of the line.
- up/down arrow key: Move up/down one line. No effect if the cursor is already at the first/last line. If the length of the previous/next line has not enough characters, then the cursor become at the end of that line.
- home/end key: Move to the beginning/end of the current line.
- mouse: No, as a hardcore programmer, Dreamoon never uses mouse when coding.

Note that the cursor in `notepad` lies either between two characters, at the beginning (before the first character), or at the end (after the last character) of a line.

Now Dreamoon has written up some code, and is going to move back and forth on it for debugging. Please help him find the fastest way to move from a curtain position to another.

Input Format

The first line of the input file contains T ($T \leq 10$), the number of test cases.

Each test case begins with a line containing n ($1 \leq n \leq 200000$), the number of lines of the code. The next line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \leq a_i \leq 10^8$), denoting the number of characters in each line of the code. The next line contains an integer q, the number of queries. Then q lines follow, each containing four integers r_1, c_1, r_2, c_2 ($1 \leq r_1, r_2 \leq n$, $0 \leq c_i \leq a_{r_i}$), indicating that Dreamoon would like to move from (r_1, c_1) to (r_2, c_2). The position (r, c) means that the cursor is after the c-th character on the r-th line. $c = 0$ stands for the beginning of the line. The total number of queries in the input does not exceed 400000.

Output Format

For each query in each test case please output one line containing one integer, indicating the minimum number of steps needed for Dreamoon to move from (r_1, c_1) to (r_2, c_2).

Sample Input

```
1
5
5 4 3 4 5
3
1 3 3 3
1 4 1 1
1 0 5 5
```

Sample Output

```
2
2
5
```